Dynamic optimization of maintenance policies for multistate system

Benoîte de Saporta, François Dufour, Huilong Zhang

Dynamic optimization of maintenance policies for multistate system

Benoîte de Saporta*, François Dufour†, Huilong Zhang‡

Abstract

This paper concerns the maintenance optimization of a multistate system subject to random failure of its components occurring with different distributions with or without aging. A dynamic maintenance policy is a sequence of condition-based intervention dates and actions. A policy generates a cost that is composed of an unavailability cost when the equipment is down or in the workshop as well as repair or replacement costs for the components. The aim of this work is to compute an approximation of the minimal mean cost over all admissible policies, and propose a policy that performs as close as possible to this minimum.

1 Introduction

This paper concerns the maintenance optimization of a multi-component equipment subject to random failure of its components occurring with different distributions with or without aging.

A dynamic maintenance policy is a sequence of (possibly time and state dependent) intervention dates and actions, where the actions can be the replacement of failed components or the repair of degraded ones and the intervention dates correspond to the dates at which these actions are performed. Preventive repair/change actions may also be taken for components in nominal state. A policy generates a cost that is composed of an unavailability cost when the equipment is down or in the workshop as well as repair or replacement costs for the components. The objective of maintenance optimization is to reduce as much as possible both the unavailability and maintenance costs in order to obtain a high level of performance at the smaller cost.

The aim of the present paper is to compute an approximation of the minimal mean cost over a wide class of non-parametric admissible policies, and propose a policy that performs as close as possible to this minimum. To to so, we model the dynamics of the equipment as Piecewise Deterministic Markov process (PDMP) Davis (1993). PDMPs form a general class of non-diffusion processes

*IMAG, Univ Montpellier, CNRS, Montpellier, France.
†INRIA CQFD, IMB, Univ Bordeaux, Bordeaux INP, CNRS, France.
‡INRIA CQFD, IMB, Univ Bordeaux, CNRS, France.

with deterministic trajectories punctuated by random jumps. As illustrated *e. g.* in Devooght (1997); Dufour and Dutuit (2002) or de Saporta et al. (2015) PDMPs are especially suitable to model dynamic reliability problems.

Our problem of maintenance optimization is an impulse control problem for PDMPs where the controller needs to choose intervention dates and actions in order to minimize some expected cost. The theory of impulse control for PDMPs is well understood and was first developed in Davis (1993). Numerical schemes to approximate the optimal performance were presented in Costa and Davis (1989) and de Saporta and Dufour (2012). In the present work, we follow the latter approach that is more suitable for nonstationary processes. We also present a completely new algorithm to approximate an optimal policy based on the theoretical work in de Saporta et al. (2017).

The remainder of this paper is organized as follows. In section 2, we present the generic model of multi-component equipment we focus on, explain how it can be modeled by a PDMP and state the optimization problem as an impulse control problem. In Section 3, we detail our discretization procedure and detail the algorithms to approximate the optimal performance and policy. In Section 4, we explain how the procedure was implemented on a specific example, how the various parameters were tuned and present our numerical results. Finally, concluding remarks are presented in Section 5.

2 Maintenance optimization problem

In this section, we state the maintenance optimization problem we aim to solve numerically. We first describe the physical system and explain how it can be modeled by a piecewise deterministic Markov process (PDMP). The maintenance optimization problem is then formalized as an impulse control problem for PDMPs.

2.1 Dynamics of the equipment

The system under consideration in this paper is a generic model for a piece of equipment with four components. Each component is subject to random deterioration and/or failure independently from the other components.

- Component 1 and 2 can be either in stable or failed state,
- Components 3 and 4 can be either in stable, degraded or failed state.

The global state of the equipment is stable if all the components are in stable state, failed if at least one component is in failed state and degraded otherwise. In other words, one can encode the state of a component as 0 corresponding to failed, 1 to degraded, and 2 to stable. Then if the state of component k is s_k, the global state of the equipment is $s = \min\{s_1, s_2, s_3, s_4\}$.

The deterioration or failure dynamics of the four components are independent and as follows, see Figure 1 for a visual summary:

- Components 1 fails according to an exponential distribution with parameter λ_1,
- Component 2 fails according to a Weibull distribution with parameters (α_2, β_2),
- Components 3 and 4 reach the degraded state according to a Weibull distribution with parameters (α_3, β_3) and (α_4, β_4) respectively, and go from degraded to failed state according to an exponential distribution with parameter λ_3 and λ_4 respectively.

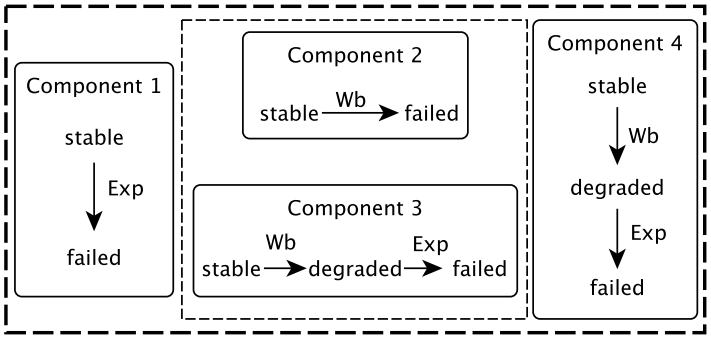

Figure 1: The equipment and possible states of its components.

The components can degrade or fail only when the equipment is functioning. The equipment can work with degraded components but is stopped at the first failure. Hence, only 16 states can be reached among the 36 possible states.

If no maintenance actions are performed, the system will thus reach the failure state and remain failed. An unavailability penalty c_u must be paid for each day the equipment spends in the failed state.

Two types of maintenance actions can be performed on the components:

- Components 1 and 2 can be changed with cost c_1 and c_2 respectively,
- Components 3 and 4 can be changed with cost c_3 and c_4 respectively or repaired with cost $r_3 < c_3$ and $r_4 < c_4$ respectively. The repair action is impossible in failed state.

Both change and repair actions are as good as new and reset the functioning time of the equipment to 0.

To perform maintenance operations, the whole equipment has to go back to the workshop and is therefore unavailable during the maintenance operations. Thus each visit to the workshop yields a fixed cost c_w in addition to the individual costs of the operations performed on each component. We will also consider that each visit to the workshop has a fixed duration T_w. We consider

that components 2 and 3 are part of a specific block: if either component needs to be changed, the whole block is changed and therefore the other component is changed too. Finally we restrict our study to the case where each visit to the workshop sets the equipment back to the stable state, which means that at least all failed components are changed and all degraded ones are repaired or changed. Stable components may also be repaired or changed. The 17 feasible combinations of interventions are listed in Table 1.

Table 1: Possible combinations of workshop operations for the equipment.

C 1	C 2	C 3	C 4
nothing	nothing	nothing	repair
nothing	nothing	repair	nothing
nothing	nothing	repair	repair
nothing	nothing	nothing	change
nothing	nothing	repair	change
nothing	change	change	nothing
nothing	change	change	repair
nothing	change	change	change
change	nothing	nothing	nothing
change	nothing	nothing	repair
change	nothing	repair	nothing
change	nothing	repair	repair
change	nothing	nothing	change
change	nothing	repair	change
change	change	change	nothing
change	change	change	repair
change	change	change	change

Recall that a maintenance policy is a sequence of intervention dates where the equipment is sent to the workshop, and repair or change actions on the components. We will only consider policies where the equipment is back in stable state after a visit to the workshop. Such a policy generates a cost corresponding to the total unavailability and maintenance costs up to a finite time horizon H corresponding to the service time of the equipment. The aim of this paper is to compute an approximation of the optimal performance, that is the minimum expected cost over all feasible maintenance policies, and to exhibit a policy that performs as close as possible to this minimum.

2.2 PDMP framework

In order to solve the optimization problem formulated above, we use the framework of impulse control for PDMPs. PDMPs form a vast class of hybrid non diffusive stochastic processes, see Davis (1993). Let M be a finite set of modes or regimes of the process. For each mode $m \in M$, denote E_m a Borel subset of $\mathbb{R}^d$. The state space of the process is then $E = \cup_{m \in M} \{m\} \times E_m$. The dynamics

is described by three local characteristics, defined as follows for $m \in M$:

- a flow $\Phi_m : E_m \times \mathbb{R}_+ \to E_m$ with a semi-group property that describes the deterministic dynamics between jumps,
- a jump intensity $\lambda_m : E_m \to \mathbb{R}_+$ that gives the frequency of the jumps,
- a Markov kernel $Q_m : E_m \times \mathcal{B}(E_m) \to E$ that selects the post-jump modes and locations.

The trajectories of the PDMP are then constructed recursively as follows. Starting from $X_0 = (m_0, x_0) \in E$, the first jump time T_1 is drawn according to the following distribution

$$\begin{aligned}&\mathbb{P}_{(m_0,x_0)}(T > t)\\ &= \mathbb{1}_{t^*_{m_0}(x_0)\geq t)} \exp\Big(- \int_0^t \lambda_{m_0}(\Phi_{m_0}(x,s)ds\Big),\end{aligned}$$

where

$$t^*_m(x) = \inf\{s > 0; \Phi_m(x,s) \in \partial E_m\}$$

is the deterministic time to reach the boundary of the state space starting from position x in mode m. On the time interval $[0, T_1)$ one has $X_t = (m_0, \Phi_{m_0}(x_0, t))$. At time T_1, a new mode m_1 and a new position x_1 are drawn according to Q: $X_{T_1} = (m_1, x_1)$. A new jump time is selected with intensity λ, and so on.

Denote T_n the n-th jump time of the process with $T_0 = 0$, $S_n = T_n - T_{n-1}$ the inter-jump times and $Z_n = X_{T_n}$ the post-jump locations. Then (Z_n, S_n) is a finite-time Markov chain that contains all the randomness of the process.

The dynamics of the equipment can be modeled by a PDMP as follows. The set M of regimes is the set of all 33 feasible combinations of the states of the components, including the workshop states, as listed in Tables **??** and 1.

For each working state m, $E_m = [0, H)^4$ where H is the time horizon of the optimization, *i.e.* the service time of the equipment. The first three coordinates correspond to the service time of components 2 to 4 respectively, and the last component is the global service time of the equipment. The flow is simply incrementing the times: for $x = (x_1, x_2, x_3, x_4)$ one has

$$\Phi_m(x, t) = (x_1 + t, x_2 + t, x_3 + t, x_4 + t).$$

The time to reach the boundary is the time left until the horizon is reached

$$t^*_m(x) = H - x_4.$$

The jump intensity is given by the appropriate sum of exponential or Weibull intensities depending on the mode, and the jump kernel determines which component degrades or fails, the new regime and functioning times being updated accordingly.

For each workshop state m, $E_m = [0, H)^4 \times [0, T_w)$. The first four coordinates are the same as above, the last one is the time spend in the workshop. As the

equipment is not functioning in the workshop, only the last two time coordinates are updated by the flow: for $x = (x_i, 1 \leq i \leq 5)$, one has

$$\Phi_m(x,t) = (x_1, x_2, x_3, x_4 + t, x_5 + t).$$

The time to reach the boundary is the time left until the horizon or the end of workshop operations is reached, whichever occurs first

$$t^*_m(x) = \min\{H - x_4, T_w - x_5\}.$$

The jump intensity is 0 as no random jump can occur. The jump kernel resets to 0 the functioning time of the components that have been repaired or changed. As we only consider policies that send the equipment to the stable state after a visit to the workshop, the new regime after exiting the workshop is always stable for all components.

Finally, we add a cemetery state Δ that the process enters when the horizon H is reached. From this date on, the flow is fixed at Δ and the jump intensity is null so that no new event can occur.

2.3 Impulse control for PDMPs

Roughly speaking, an impulse control problem corresponds to choosing the best intervention dates and new mode and location after the interventions in order to minimize some cost. More precisely, a policy $\mathcal{S}$ is a sequence $(\tau_n, R_n)_{n\geq 1}$ where τ_n are the controller-chosen intervention dates and R_n are the controller-chosen new mode and position of the process after the intervention. In our maintenance optimization context, the intervention dates are the dates when the controller decides to send the equipment to the workshop and the new mode and position correspond to the specific maintenance actions performed on each component.

The cost of policy $\mathcal{S}$ when the process starts from (m_0, x_0) is defined as

$$\begin{aligned}\mathcal{J}^{\mathcal{S}}(m_0,x_0) =& \mathbb{E}^{\mathcal{S}}_{(m_0,x_0)}\Big[\int_0^\infty e^{-\alpha s} f(X_s)ds \\ &+ \sum_{i=1}^{\infty} e^{-\alpha\tau_i} c(X_{\tau_i}, X_{\tau_i^+})\Big],\end{aligned}$$

where f is the running cost, c the cost of impulsions and α a discount factor that will be set to 0 in the sequel. In our maintenance example, f equals 0 in stable, degraded or workshop regimes, and $f = c_u$ the unavailability penalty in failed regimes. The cost of impulsion depends on the nature of the operations performed, it is the suitable combination of repair and/or change costs for the components added to the fixed unavailability cost c_w.

The value function of the impulse control problem is the minimum cost over the set $\mathbb{S}$ of all admissible policies

$$\mathcal{V}(x) = \inf_{\mathcal{S}\in\mathbb{S}} \mathcal{J}^{\mathcal{S}}(x).$$

The precise definition of $\mathbb{S}$ is rather technical and omitted here. Most importantly, the intervention times must be non-anticipative. The interested reader may consult Costa and Davis (1989) for details. Our aim is to compute a numerical approximation of $\mathcal{V}$ when interventions are only possible in the degraded or failed states. Although it is relevant to evaluate the value function, as it is the best possible performance, the most important for the controller is to find a feasible strategy that performs as close as possible to this optimum. Constructing such a strategy is the other aim of this paper.

3 Numerical approximation

As proved in Costa and Davis (1989), the value function is the fixed point of some functional operator. Hence, a strategy to approximate $\mathcal{V}$ is to discretize this operator. This is the approach developed in de Saporta and Dufour (2012) that we will implement here. Its main strong point is that it is based on simulations of the non-controlled process only.

3.1 Discretization of the control set

In order to apply the method developed in de Saporta and Dufour (2012), we need a finite control set. The control set $\mathbb{U}$ is the set of all modes and positions the process can restart from after an intervention. In our case study, an intervention corresponds to sending the equipment to the workshop. The number of workshop modes is finite, but the functioning times of the various components and equipment are not. We choose to discretize possible times on a finite cartesian grid of the space of feasible points

$$\{x \in [0, H)^4; \max\{x_1, x_2, x_3\} \leq x_4\},$$

as no component can work longer than the whole equipment. The lag of the grid is chosen empirically, see Section 4. We thus obtain a finite control set denoted U, with cardinal u.

3.2 Discretizations of the Markov chain (Z_n, S_n)

The main ingredient of our approximation is a time-dependent discretization of the state space based on the optimal quantization of the post-jump location – inter-jump times Markov chain (Z_n, S_n). Optimal quantization is a simulation-based method to approximate a continuous state space Markov chain by a finite state space one, see Pagès et al. (2004). Using Algorithms 1 and 2, one obtains a finite grid Γ_n at each date n as well as transition matrices P_n from grid Γ_{n-1} to grid Γ_n.

Such grids strongly depend on the initial distribution of Z_0 as our process is far from stationary. We will use several sequences of grids. We denote Γ_n the grids with starting point $Z_0 = (m_0, x_0)$ and $S_0 = 0$ where m_0 is the stable state and $x_0 = (0, 0, 0, 0)$. We denote Γ_n^y the grids with starting point $Z_0 = y$ and

Algorithm 1: CLVQ algorithm to quantize the Markov chain (Z_n, S_n)

input : Time horizon N, Number of points K in each grid, Initial grids (Γ_n^0) with K points, Number of runs N_R, simulator of target Markov chain (Z_n, S_n)
output: Optimized grids (Γ_n), $0 \leq n \leq N$

begin
  for $k \leftarrow 0$ *to* $N_R - 1$ **do**
    draw trajectory $(z_n, s_n)_{0 \leq n \leq N}$ of Markov chain $(Z_n, S_n)_{0 \leq n \leq N}$
    for $n \leftarrow 0$ *to* N **do**
      select ζ closest neighbor of (z_n, s_n) in Γ_n^k
      $\zeta' \leftarrow \zeta - \frac{4K^{1/2}}{4K^{1/2} + \pi^2 k/K} \|\zeta - (z_n, s_n)\|$
      $\Gamma_n^{k+1} \leftarrow \Gamma_n^k \cup \{\zeta'\} \backslash \{\zeta\}$
    end
  end
  return : $(\Gamma_n^{N_R})$, $0 \leq n \leq N$
end

Algorithm 2: Computation of the transition matrices P_n

input : Time horizon N, Quantization grids (Γ_n) with K points, $0 \leq n \leq N$, number N_{MC} of Monte Carlo runs
output: Transition probabilities $P_n(i,j)$, $1 \leq n \leq N$, $1 \leq i, j \leq K$

begin
  count(n,i,j)$\leftarrow$0, $1 \leq n \leq N$, $1 \leq i, j \leq K$
  for $k \leftarrow 1$ *to* N_{MC} **do**
    draw trajectory $(z_n, s_n)_{0 \leq n \leq N}$ of $(Z_n, S_n)_{0 \leq n \leq N}$
    for $n \leftarrow 0$ *to* $N - 1$ **do**
      select $i \leq K$ such that ζ_n^i be the closest neighbor of (z_n, s_n) in Γ_n
      select $j \leq K$ such that ζ_{n+1}^j be the closest neighbor of (z_{n+1}, s_{n+1}) in Γ_{n+1}
      count(n+1,i,j)$\leftarrow$ count(n+1,i,j)+1
    end
  end
  $P_n(i,j) \leftarrow$count(n,i,j)$/N_{MC}$, $1 \leq n \leq N$, $1 \leq i, j \leq K$
  return : $P_n(i,j)$, $0 \leq n \leq N-1$, $1 \leq i, j \leq K$
end

$S_0 = 0$ for each y in the finite control set U. We thus have $u+1$ sequences of quantization grids.

It is important to notice that although we use different starting points, we always quantize the non-controlled process. Hence, starting from the stable state and initial functioning times set at 0, there is a maximum of $N = 4$ jumps before reaching the cemetery state (two degradations, then failure and cemetery). Starting from y in the control set U, the maximum is $N = 5$ jumps (one jump to go from the workshop to the stable state, then two degradations, failure and cemetery). We thus need to compute $5 + 6(u+1)$ quantization grids and $4 + 5(u+1)$ transition matrices.

3.3 Approximation of the value function

We then proceed to construct the approximate value function following the procedure described in de Saporta and Dufour (2012). For self-containedness, we recall the two steps of the procedure in Algorithms 3 and 4. The approximate value function at the starting point $Z_0 = (m_0, x_0)$ and $S_0 = 0$ where m_0 is the stable state and $x_0 = (0, 0, 0, 0)$ is the output $\hat{v}_0$ of Algorithm 4. Note that both algorithms have extra lines (28 to 30 in Algorithm 3, 15 to 17 in Algorithm 4) corresponding to the computations of the outputs win_1, win_2 and s^* that will be useful in the next section for the computation of a strategy close to optimality.

These algorithms depend on a number of parameters. The number of iterations N_{it} should be large as the theoretical value function is obtained with an infinite number of iteration of the underlying operator. The bounding function g must be greater than the average cost of the no-impulse strategy. In our case, one has, for $(m, x) \in E$,

$$\begin{aligned} g(m,x) &\geq \mathbb{E}_{(m,x)}\Big[\int_0^\infty f(X_s)ds\Big] \\ &= c_u \mathbb{E}_{(m,x)}[T_f], \end{aligned}$$

where T_f is the time spent in the failed state. For instance, one can take the constant function $g(m, x) = c_u H$ or $g(m, x) = c_u(H - x_4)$ with a slightly refined evaluation of the time left until the horizon. The discretization step δ should be small, and can be state dependent instead of being uniform. Finally, the function F is defined as follows

$$F(m,x,t) = \mathbb{E}_{(m,x)}\Big[\int_0^{T_1 \wedge t} f(\Phi(x,s))ds\Big],$$

so that in our case $F = 0$ in the stable, degraded, cemetery and workshop modes, and for the failure mode

$$F(m,x,t) = c_u(t \wedge (H - x_4)).$$

Algorithm 3: Approximation of the value functions on the control grid U

input : Grids $(\Gamma_n^y)_{0\le n\le 5, y\in U}$, Transition matrices $(P_n^y)_{1\le n\le 5, y\in U}$, bounding function g, discretization step δ, number of iterations $N_{it}\ge 5$

output : $\widetilde{v}_n(y)$, $win_1(z,n,k,y)$, $win_2(z,n,k,y)$, $s^*(z,n,k,y)$, $y\in U$, $z\in\Gamma_k^y$, $1\le k\le 5$, $1\le n\le N_{it}-1$

begin
  for $y\in U$ **do**
    $\widetilde{v}_{N_{it}}(y)\leftarrow g(y)$
  end
  for $n\leftarrow N_{it}-1$ **to** 1 **by step:** -1
  do
    for $y\in U$ **do**
      if $N_{it}-n\ge 6$ **then**
        $\widehat{v}^n_{N_{it}}\leftarrow 0$
      else
        for $(z,s)\in\Gamma^y_{N_{it}-n}$ **do**
          $\widehat{v}^n_{N_{it}}(z)\leftarrow g(z)$
        end
      end
      for $k\leftarrow N_{it}-n-1$ **to** 0 **by step:** -1
      do
        if $k\ge 6$ **then**
          $\widehat{v}^n_{n+k}\leftarrow 0$
        else
          for $(z,s)\in\Gamma_k^y$ **do**
            $\widehat{K}(z)\leftarrow \sum_{(z',s')\in\Gamma^y_{k+1}} P^y_{k+1}(z,(z',s'))\big(\widehat{v}^n_{n+k+1}(z')+F(z,t^*(z))\big)$
            for $t\leftarrow 0$ **to** $t^*(z)-\delta$ **by step:** δ
            do
              $w(z,t)=\min_{y'\in U}\{c(\Phi(z,t),y')+\widetilde{v}_{n+1+k}(y')\}$
              $\widehat{J}(z,t)\leftarrow \sum_{(z',s')\in\Gamma^y_{k+1}} P^y_{k+1}(z,(z',s'))\big(F(z,t)+\widehat{v}^n_{n+k+1}(z')\mathbb{1}_{\{s'<t\}}+ w(z,t)\mathbb{1}_{\{s'\ge t\}}\big)$
            end
            $\widehat{v}^n_{n+k}(z)\leftarrow \min_t\{\widehat{J}(z,t)\}\wedge\widehat{K}(z)\}$
            $win_1(z,n,k,y)\leftarrow(\min_t\{\widehat{J}(z,t)\}\le\widehat{K}(z)$
            $win_2(z,n,k,y)\leftarrow\arg\min_{y'\in U}\{c(\Phi(z,t),y')+\widetilde{v}_{n+1+k}(y')\}$
            $s^*(z,n,k,y)\leftarrow\arg\min_t\{\widehat{J}(z,t)\}$
          end
        end
        $\widetilde{v}_n(y)\leftarrow\widehat{v}^n_n(y)$
        return : $\widetilde{v}_n(y)$
      end
    end
  end
end

Algorithm 4: Approximation of the value function at the point (m_0, x_0) where m_0 is the stable state and $x_0 = (0,0,0,0)$.

input : Grids $(\Gamma_n)_{0\le n\le 4}$, Transition matrices $(P_n)_{1\le n\le 4}$, bounding function g, time discretization step δ, $(\widetilde{v}_n(y))_{1\le n\le N_{it}, y\in U}$ output of Algorithm 3

output : $\widehat{v}_0$, $win_1(z,k)$, $win_2(z,k)$, $s^*(z,k)$, $y \in U$, $z \in \Gamma_k$, $0 \le k \le 4$

begin

  for $(z,s) \in \Gamma_4$ **do**

    $\widehat{v}_5(z) \leftarrow g(z)$

  end

  for $k \leftarrow 4$ **to** 0 **by step:** -1

  do

    for $(z,s) \in \Gamma_k$ **do**

      $\widehat{K}(z) \leftarrow \sum_{(z',s')\in\Gamma_{k+1}} P_{k+1}(z,(z',s'))\big(\widehat{v}_{k+1}(z') + F(z,t^*(z))\big)$

      for $t \leftarrow 0$ **to** $t^*(z) - \delta$ **by step:** δ

      do

        $w(z,t) = \min_{y\in U}\{c(\Phi(z,t),y) + \widetilde{v}_{k+1}(y)\}$

        $\widehat{J}(z,t) \leftarrow \sum_{(z',s')\in\Gamma_{k+1}} P_{k+1}(z,(z',s'))\big(F(z,t) + \widehat{v}_{k+1}(z')\mathbb{1}_{\{s'<t\}} + w(z,t)\mathbb{1}_{\{s'\ge t\}}\big)$

      end

      $\widehat{v}_k(z) \leftarrow \min_t\{\widehat{J}(z,t)\} \wedge \widehat{K}(z)\}$

      $win_1(z,k) \leftarrow (\min_t\{\widehat{J}(z,t)\} \le \widehat{K}(z)$

      $win_2(z,k) \leftarrow \arg\min_{y\in U}\{c(\Phi(z,t),y) + \widetilde{v}_{k+1}(y)\}$

      $s^*(z,k) \leftarrow \arg\min_t\{\widehat{J}(z,t)\}$

    end

  end

  return : $\widehat{v}_0$

end

3.4 Construction of a feasible policy

We now present a completely new procedure to determine an explicit policy close to optimality. Algorithm 5 explains how to simulate a trajectory according to this policy.

4 Numerical results

The main technical difficulty consists in tuning the various discretization parameters in order to obtain an acceptable compromise between precision and complexity. To do so, we selected some reference policies and computed their exact and approximated costs to empirically evaluate the approximation error. Note that a theoretical error bound is provided in de Saporta and Dufour (2012) but it is too conservative to be used in practice for parameter tuning. All steps of the implementation were validated through a sensitivity analysis with different starting points for the process.

The numerical values of the equipment are set to arbitrary values so that the cost of the no-impulse policy for the exact simulator over an horizon of $H = 10^4$ time units is 20000.

4.1 Reference policies

We used the two following reference policies.
Policy A is a corrective maintenance policy where the equipment is sent back to the workshop after 1 unit of time spent in the failed state. All failed components are changed, and all degraded components are repaired.
Policy B is a preventive maintenance policy where the equipment is sent back to the workshop after 1 unit of time spent in the degraded state. All failed components are changed, and all degraded components are repaired.
Both policies can be exactly simulated. Their cost was evaluated through 10^6 Monte Carlo simulations and are given in Table 2. As expected, Policy A performs significantly better than the no-impulse policy and Policy B performs better than policy A.

4.2 Discretization parameters

To tune the number of points in the discretized control set U, we compared the average cost of Policy B with the continuous control set $\mathbb{U}$ and with projecting the functioning times of the equipment on a discretized set at each impulsion. We tried a high number of combinations summarized in Table 3. The errors are expressed as relative errors with respect to the exact cost of Policy B. We chose the grid with 2195 points yielding a relative error of 6% as an acceptable compromise between precision and complexity.

Regarding the number of points K in the quantization grids, we compared again the exact and approximate costs of the reference policies for K between 10

Algorithm 5: Simulation of an optimally controlled trajectory

input : Grids $(\Gamma_n^y)_{0\leq n\leq 5, y\in U}$, Grids $(\Gamma_n)_{0\leq n\leq 4}$, outputs of Algorithms 3 and 4
output : Trajectory $(Z_j, S_j)_{j\geq 0}$

begin
  $j \leftarrow 0$
  $z \leftarrow (stable, stable, stable, stable, 0, 0, 0, 0)$
  $s \leftarrow 0$
  $(Z_j, S_j) \leftarrow (z, s)$
  $jump \leftarrow 0$
  $impulse \leftarrow 0$
  while $j < 4$ *and* $impulse = 0$ *and* $(z, s) \neq \Delta$ **do**
    $(\widehat{z}, \widehat{s}) \leftarrow proj_{\Gamma_j}(z, s)$
    $s'' \leftarrow s^*(\widehat{z}, j)$
    Draw the next post-jump location and inter-jump time (z', s') starting from z
    if $win_1 = 0$ *or (*$win_1 = 1$ *and* $s'' \geq s'$*)* **then**
      $jump \leftarrow jump + 1$
      $(z, s) \leftarrow (z', s')$
    else
      $impulse \leftarrow impulse + 1$
      $y \leftarrow win_2(\widehat{z}, j)$
      $(z, s) \leftarrow (y, s'')$
      $n \leftarrow jump + impulse$
      $k \leftarrow 0$
    end
    $j \leftarrow j + 1$
    $(Z_j, S_j) \leftarrow (z, s)$
  end
  while $impulse > 0$ *and* $jump + impulse \leq N_{it}$ *and* $(z, s) \neq \Delta$ **do**
    $(\widehat{z}, \widehat{s}) \leftarrow proj_{\Gamma_k^y}(z, s)$
    $s'' \leftarrow win_2(\widehat{z}, k, n, y)$
    Draw the next post-jump location and inter-jump time (z', s') starting from z
    if $win_1(\widehat{z}, k, n, i) = 0$ *or (*$win_1(\widehat{z}, k, n, i) = 1$ *and* $s'' \geq s'$*)* **then**
      $jump \leftarrow jump + 1$
      $k \leftarrow k + 1$
      $(z, s) \leftarrow (z', s')$
    else
      $impulse \leftarrow impulse + 1$
      $y \leftarrow win_2(\widehat{z}, k, n, y)$
      $(z, s) \leftarrow (y, s'')$
      $n \leftarrow jump + impulse$
      $k \leftarrow 0$
    end
    $j \leftarrow j + 1$
    $(Z_j, S_j) \leftarrow (z, s)$
  end
  return : $(Z_j, S_j)_{j\geq 0}$
end

Table 2: Costs of the reference policies.

No-impulse	Policy A	Policy B
20000	11295	8495

Table 3: Relative errors due to the discretization of the control set for Policy B.

Cardinality of cartesian grid	Number of feasible points	relative error
$3 \times 3 \times 3 \times 5$	419	0.1458
$4 \times 4 \times 4 \times 5$	627	0.1331
$5 \times 5 \times 5 \times 5$	1055	0.1235
$3 \times 3 \times 3 \times 11$	788	0.0962
$4 \times 4 \times 4 \times 11$	1219	0.0819
$5 \times 5 \times 5 \times 11$	1855	0.0730
$6 \times 6 \times 6 \times 11$	2790	0.0672
$7 \times 7 \times 7 \times 11$	3570	0.0634
$8 \times 8 \times 8 \times 11$	4647	0.0604
$3 \times 3 \times 3 \times 21$	1403	0.0775
$4 \times 4 \times 4 \times 21$	2195	0.0626
$5 \times 5 \times 5 \times 21$	3423	0.0534
$6 \times 6 \times 6 \times 21$	4900	0.0436
$7 \times 7 \times 7 \times 21$	6489	0.0384
$8 \times 8 \times 8 \times 21$	8399	0.0350

and 1000. We chose $K = 50$ yielding a 0.3% relative error on the cost of Policy B.

4.3 Approximate value function

We can now turn to the approximation of the value function. It represents the best possible performance when allowing all 17 maintenance actions at any time in the degraded or failed states. Regarding the number of iterations N_{it} for Algorithm 3, we tried all values between 6 and 19. The resulting approximated value function are represented on Figure 2. Convergence seems to be reached for $N_{it} = 13$. This yields a relative gain with respect to the no-impulse policy of 65.9% and a relative gain of 19.6% with respect to Policy B. This gain is significant compared to the empirical approximation error.

As there is a significant gain between the value function and the best reference policy, it is crucial to be able to compute a policy and corresponding decision rules that yield this reduced cost. The implementation of Algorithm 5 is still on-going. Results will be presented at the conference.

5 Conclusion

We have studied a problem of maintenance optimization for a multi-component equipment. The dynamics of the equipment has been modeled by a continuous-time piecewise deterministic Markov process. The optimization problem has

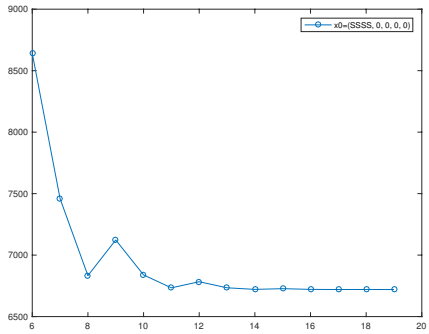

Figure 2: Approximated value function for several numbers of iterations

been translated into an impulse control problem for the PDMP. We have implemented an algorithm to approximate the value function, that approximated the best possible performance over a wide (non parametric) set of maintenance policies. The main advantage of this algorithm is that it is based on simulations of the non-controlled process only. The main drawback is the number of parameters to be empirically tuned and the high computation time.

We obtained that the optimal performance is significantly better than the best preventive reference policy. This is probably due to the fact that we allow interventions even on the components in the stable state. This approach also allows to obtain an explicit policy close to optimality. This part is still under implementation, and it should be very interesting to understand how this policy differs from the reference ones.

Acknowledgement

This work benefited from the support of the FMJH Program PGMO and from the support of EDF and Thales

References

Costa, O. L. V. and M. H. A. Davis (1989). Impulse control of piecewise-deterministic processes. *Math. Control Signals Systems 2*(3), 187–206.

Davis, M. (1993). *Markov models and optimization*, Volume 49 of *Monographs on Statistics and Applied Probability*. Chapman & Hall, London.

de Saporta, B. and F. Dufour (2012). Numerical method for impulse control of piecewise deterministic markov processes. *Automatica 48*, 779–793.

de Saporta, B., F. Dufour, and A. Geeraert (2017). Optimal strategies for impulse control of piecewise deterministic markov processes. *Automatica J. IFAC 77*, 219–229. arXiv:1603.07859, hal-01294286.

de Saporta, B., F. Dufour, and H. Zhang (2015). *Numerical methods for simulation and optimization of piecewise deterministic Markov processes: application to reliability*. Mathematics and statistics series. Wiley-ISTE. hal-01249897.

Devooght, J. (1997). *Dynamic reliability*. Advances in nuclear science and technology. Berlin: Chapman and Hall.

Dufour, F. and Y. Dutuit (2002). Dynamic reliability : A new model. In *Proceedings of ESREL 2002 Lambda-Mu 13 Conference*, pp. 350–353.

Pagès, G., H. Pham, and J. Printems (2004). An optimal Markovian quantization algorithm for multi-dimensional stochastic control problems. *Stoch. Dyn. 4*(4), 501–545.